Runtime randomization and perturbation for virtual machines.

JAVIER CABRERA ARTEAGA

Licentiate Thesis in [Research Subject - as it is in your ISP]
School of Information and Communication Technology
KTH Royal Institute of Technology
Stockholm, Sweden [2022]

KTH School of Information and
Communication Technology
SE-164 40 Kista
SWEDEN

TRITA-ICT XXXX:XX
ISBN XXX-XX-XXXX-XXX-X

Akademisk avhandling som med tillstånd av Kungl Tekniska högskolan framläges till offentlig granskning för avläggande av licentiatexamen i [ämne/subject] [veckodag/weekday] den [dag/day] [månad/month] [år/2022] klockan [tid/time] i [sal/hall], Electrum, Kungl Tekniska högskolan, Kistagången 16, Kista.

Tryck: Universitetsservice US AB

Abstract

Write your abstract here...
Keywords: Keyword1, keyword2, ...

Sammanfattning

Write your Swedish summary (popular description) here...
Keywords: Keyword1, keyword2, ...

Acknowledgements

Write your professional acknowledgements here...

Acknowledgements are used to thank all persons who have helped in carrying out the research and to the research organizations/institutions and/or companies for funding the research.

Name Surname,
Place, Date

Contents

Contents **vi**

List of Figures **viii**

List of Tables **ix**

List of Acronyms **xi**

1 Introduction **1**
1.1 Motivation 1
1.1.1 Why variants ? 1
1.1.2 Research questions 1
1.2 Contributions 1

2 Background and State of the art **3**
2.1 CROW 3

3 Methodology **5**
3.1 Corpora 5
3.2 RQ1. To what extent can we generate program variants for WebAssembly? 7
3.3 RQ2. To what extent are the generated variants dynamically different? 9
3.4 RQ3. To what extent the artificial variants exhibit different execution times on Edge-Cloud platforms? 11
3.5 Conclusions 13

4 Results **15**
4.1 RQ1. To what extent can we generate program variants for WebAssembly? 15
4.2 Answer to RQ1. 17
4.3 RQ2. To what extent are the generated variants dynamically different? 17
4.4 Answer to RQ2. 21

4.5 RQ3. To what extent the artificial variants exhibit different execution times on Edge-Cloud platforms? 21
4.6 Answer to RQ3. 22
4.7 Conclusions 23

5 Conclusions **25**

List of Figures

2.1 CROW workflow to generate program variants. CROW takes C/C++ source codes or LLVM bitcodes to look for code blocks that can be replaced by semantically equivalent code and generates program variants by combining them. 3

3.1 The program variants generation for RQ1. 8
3.2 Dynamic analysis for RQ2. 10
3.3 Multivariant binary creation and workflow for RQ3. 12

4.1 Pairwise comparison of programs' population traces in logarithmic scale. Each vertical group of blue dots represents a programs' population. Each dot represents a comparison between two program execution traces according to Metric 2. 19
4.2 Execution time distributions for `Hilber_curve` program and its variants. Baseline execution time mean is highlighted with the magenta horizontal line. 20
4.3 Execution time distributions. Each subplot represents the quantile-quantile plot of the two distributions, original and multivariant binary. 22

List of Tables

3.1 Corpora description. The table is composed by the name of the corpus, the number of modules, the number of programs, the number of functions, the lines of code range and the location of the corpus. 7

4.1 General diversification results. The table is composed by the name of the corpus, the number of functions, the number of succesfully diversified functions, the number of non-diversified functions and the cumulative number of variants. 16

List of Acronyms

Wasm	WebAssembly
DTW	Dynamic Time Warping